



REC'D PCT/PTO 27 SEP 2003



PCT/IB U3/00.626

14.02.03

10/509220



INVESTOR IN PEOPLE

REC'D · 27 FEB 2003	
WIPO	PCT

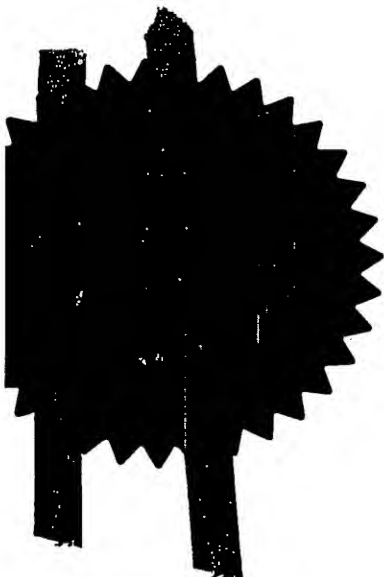
The Patent Office  
Concept House  
Cardiff Road  
Newport  
South Wales  
NP10 8QQ

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.



Signed

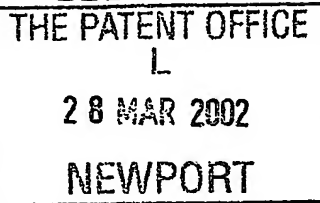
Dated 18 November 2002

**BEST AVAILABLE COPY**

**PRIORITY  
DOCUMENT**  
SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH RULE 17.1(a) OR (b)

**Request for grant of a patent**

(See notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form)



**The Patent Office**  
Cardiff Road  
Newport  
Gwent NP10 8QQ

Your reference

PHGB020032

Patent application number  
(The Patent Office will fill in this)

0207296.5

28MAR02 E707210-2 002879  
P01/7700 0.00-0207296.5

28 MAR 2002

Full name, address and postcode of the or of each applicant (*underline all surnames*)

KONINKLIJKE PHILIPS ELECTRONICS N.V.  
GROENEWOUDSEWEG 1  
5621 BA EINDHOVEN  
THE NETHERLANDS

Patents ADP Number (*if you know it*)

741929400

If the applicant is a corporate body, give the country/state of its incorporation

THE NETHERLANDS

Title of the invention

METHOD AND APPARTUS FOR CONTEXT SWITCHING  
IN COMPUTER OPERATING SYSTEMS

Name of your agent (*if you have one*)  
"Address for service" in the United Kingdom to which all correspondence should be sent (*including the postcode*)

STEVE TOWNSEND  
Philips Intellectual Property and Standards  
Cross Oak Lane  
Redhill  
Surrey RH1 5HA

Patents ADP number (*if you know it*)

8226961001

If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and (*if you know it*) the or each application number

Country

Priority Application number  
(*if you know it*)

Date of filing  
(*day/month/year*)

If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application

Number of earlier application

Date of filing  
(*day/month/year*)

Is a statement of inventorship and of right to grant of a patent required in support of this request? (*Answer "Yes" if:*

YES

- a) any applicant named in part 3 is not an inventor, or
  - b) there is an inventor who is not named as an applicant, or
  - c) any named applicant is a corporate body.
- See note (d))

2. Enter the number of sheets for any of the following items you are filing with this form.  
Do not count copies of the same document.

Continuation sheets of this form

Description	6
Claims(s)	2
Abstract	1
Drawings	3

*DMC only*

0. If you are also filing any of the following, state how many against each item:

Priority Documents

Translations of priority documents  
Statement of inventorship and right  
to grant of a patent (*Patents Form 7/77*)  
Request for preliminary examination and  
search (*Patents Form 9/77*)  
Request for substantive examination  
(*Patents Form 10/77*)  
Any other documents  
(*Please specify*)

1. I/We request the grant of a patent on the basis of this application.

Signature

*Stomsend*

Date

*27/03/02*

2. Name and daytime telephone number of person to contact in the United Kingdom

01293 815339

(S. Townsend)

**Warning**

After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.

**Notes**

If you need help to fill in this form or you have any questions, please contact the Patent Office on 0645 500505.

Write your answers in capital letters using black ink or you may type them.

If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.

If you have answered "Yes" Patents Form 7/77 will need to be filed.

Once you have filled in the form you must remember to sign and date it.

For details of the fee and ways to pay please contact the Patent Office.

## DESCRIPTION

**METHOD AND APPARATUS FOR CONTEXT SWITCHING  
IN COMPUTER OPERATING SYSTEMS**

5

The present invention relates to a method and apparatus for context switching in computer operating systems.

Operating systems are used within computer systems so as to increase  
10 the utilisation of the system's processor. In view of the difference in operating speed between the processor and, for example input output devices, the operating system advantageously serves to schedule instructions to the processor in order to limit the time period for which the processor might otherwise remain idle due to its faster operating speed.

15 With developments in multitasking, and multithreading, an increasing number of interruptions are required to be handled by the operating system and correspondingly frequent context switches between different processes are then required.

With, for example, UNIX-like kernels, the switching between running  
20 processes should be as fast as possible and employ the minimum possible number of processor cycles. However, before a context switch to a new process can be achieved, the cached data of the previously running process may be written back to memory for consistency between the contents of the cache and memory. The time taken to write the cached data back to memory  
25 once it has been identified that a context switch is to be conducted is a disadvantageously limiting feature with regard to the overall time required to execute the context switch.

Attempts have been made to reduce context switching time as  
illustrated for example in US-A-6 006 320. Here a duplicate arithmetic logic  
30 unit, instruction and data cache, register set and load/store unit are provided in an attempt to speed up the context switching process. However, such an approach to this problem can be considered disadvantageous in requiring the

aforementioned duplication of hardware and proves to be unnecessarily complex.

5 The present invention therefore seeks to provide for a simplified solution to the problem of context switching time delays.

According to one aspect of the present invention there is provided a method of context switching between processes in a computer operating system including writing cached data back to a memory means, comprising the  
10 step of the writing cached data back to the memory means during processor idle cycles at completion of a process and prior to initiation of the context switch.

The invention is advantageous in that it employs the idle loop entered by the processor once, for example, a runnable process has completed and  
15 while the processor is waiting for the next event, such as the completion of an input/output operation, to occur. During such an idle loop, spare processor cycles are lost and the present invention advantageously makes use of such cycles in order to write-back the cached data to memory.

Thus, at the end of a process, cached data can somewhat automatically  
20 be written back to memory so that, should a context switch be required, there is then no need to first write back the cached data to memory before commencing with the context switch. This can increase substantially the switching time required for context switches in an operating system.

The feature of Claim 2 is advantageous in providing for an effective  
25 means for indicating that cached data write-back has occurred so as to lead to appropriate subsequent control steps depending on whether a context switch is required or not.

The subject matter of Claims 3-5 advantageously further adapts the subject matter of Claim 2 in a particularly effective manner so as to lead to a  
30 suitable form of process switching as required.

According to another aspect of the present invention, there is provided a computer implemented system including a processor and cache memory

arranged to receive operating system instructions for context switching between processes, and including control means for writing back cached data to memory means during processor idle cycles at completion of a process, and prior to initiation of the context switch.

5 In accordance with a further aspect of the present invention, there is provided a computer program product having computer program instructions and arranged for controlling context switching between processes in a computer operating system so as to write cached data back to memory means during processor idle cycles at completion of a process and prior to initiation of  
10 the context switch.

As will be appreciated the present invention can therefore be provided as a software, or hardware, solution to the problem discussed above.

The present invention is described further hereinafter, by way of  
15 example only, with reference to the accompanying drawings in which:

Fig. 1 is a flow diagram of a context switching operation according to an embodiment of the present invention;

Fig. 2 is a schematic block diagram of a computer system including an operating system and related control means according to an embodiment of  
20 the present invention; and

Fig. 3 is a schematic block diagram of a computer system embodying the present invention and illustrating the transfer of cached pages.

As will be appreciated, the invention proposes the use of idling time to  
25 write the cached data back to memory of the last running process, hence removing the time penalty of this operation at process switching time. Once the cached data has been written back to memory, a flag for that process is set to indicate that it has been done.

When a switch between runnable processes occurs, the last runnable  
30 processes's flag is checked to see if its cached data has been written back during a CPU idle, and if set, the switch does not need to write cache back. However, if the CPU idle has not been entered, and hence the last runnable

processes's flag is not set, then the process switching operating need to write the cache back.

The resulting advantage is that for a system that has free CPU idle time, switching between processes is faster and the system becomes more responsive to the user. With current fast processors, processor idle time is available for most moderately loaded systems. Processor idle time also occurs when processes are using Input/Output steps and the processor is waiting for these transactions to complete.

Turning first to Fig.1, there is provided a flow diagram illustrating an embodiment of the present invention in which a current process 10 is to be rescheduled 12. At step 14 a search for the next runnable process is made and, if one is located, a check at 16 is made to ascertain if the previous running process's flag has been set. If no flag has been set then a decision is made at 18 to copy back the cache of the previous running process and the flag of the next runnable process identified at 14 is cleared at 20. If at 14 it was determined that the previous running process's flag had been set, then the method skips on to block 20 directly. At 22 the context switch to the new process is made and, at 24, the new process becomes the currently running process.

Back at block 14, should a next runnable process not have been found, then the method checks to see if the previous running process's flag has been set. If the flag has not been set, then the cached data is copied back at 28 and a flag in the previous running process's process descriptor is set at 30. A sleep mode is then entered at 32 until, for example, an interrupt occurs and the method returns to block 14 as part of an idle loop.

If at block 26 it is determined that the previous running process's flag has been set, then the method skips to block 30 as illustrated.

Fig. 2 illustrates computer system 34 embodying the present invention and which comprises the operating system kernel 36 and user mode processes 38. The user mode processes comprises three processes 40, 42 and 44 and the system illustrates the process 40 being blocked such that a scheduler 44 reschedules to process 42. Each of the runnable processes is

associated with a respective runnable process list 46, 48 having respective flags 46a, 48a. The configuration illustrated in Fig. 2 is such as to illustrate a context switch between processes and the manner in which the runnable process descriptor lists point to identify the runnable processes.

5        Lastly, Fig. 3 illustrates a CPU 50, data cache 52 and memory 54 of a computer system 56, and illustrates in particular the manner in which cached pages 58 – 62 are written back to respective memory locations 64 – 68 of the memory in accordance with the present invention. As will be appreciated, the data cache pages 58 - 62 are written back to the memory 54 if the process's  
10    flag is not set.

        Analysis on a Philips TriMedia 1300 (with an external memory manager unit) prototype board has shown that a typical 1Mb user mode application takes about 6000-7000 CPU cycles to write the process's data cache back to memory, which is about 30-50% of the total context switch time. With the data  
15    cache being written to memory during CPU idle time, the context switch time loses this 30-50% overhead and so almost doubles the context switch speed. It will of course be appreciated that these figures vary, depending on the size of the application and the number of cached regions of memory.

        The present invention finds particular use in UNIX-like systems, such as  
20    Linux, FreeBSD, Solaris, etc and that use a processor with a level-1 or level-2 cache. All such UNIX-like systems have an idle loop, or idle process, that employ effectively wasted CPU cycles, and most modern processors have some form of memory cache in, or between, the processor and main memory.

        The present invention can therefore find use in most current computer  
25    operating systems.

        As noted before, the invention can comprise a software, or hardware, solution to the problem of the prior art. If the idea is implemented in hardware, one of a variety of methods could be used. For example, the CPU sleep mode could be employed for the data cache write-back automatically during a sleep  
30    period. Also, "flush data cache during sleep" flag in a status register could be employed to inform the CPU sleep operation to conduct the data cache write-back. Further, an extra sleep op-code could be added that performs the data



cache write-back, in addition to the normal sleep op-code. The addition of a data cache write-back operand to a CPU sleep opcode is also an option.

Thus, as will be appreciated, the present invention advantageously employs processor idle time to accelerate, and simplify, context switching in  
5 computer operating systems. The invention overcomes the problem of relatively slow context switching speeds between runnable processes/tasks which arises due to the general requirement to write cached data back to memory. Since, in the prior-art, this writing-back of cached memory is carried out at the time of the actual context switch, the employment of otherwise spare  
10 processor cycles in the idle loop of the operating system advantageously serves to enhance the operating speed since, at the time of initiating the context switch, the cached data has already been written back to memory.

## CLAIMS

1. A method of context switching between processes in a computer operating system including writing cached data back to a memory means, comprising the step of the writing cached data back to the memory means  
5 during processor idle cycles at completion of a process and prior to initiation of the context switch.

2. A method as claimed in Claim 1, including the step of setting a flag to indicate cached data has been written back to memory subsequent to  
10 the completion of the process.

3. A method as claimed in Claim 2, including the step of, at the time of requiring a context switch, checking for the said flag to identify if the previous process's cached data has been written back to memory.  
15

4. A method as claimed in Claim 3, including the step of, upon identifying that the said flag has been set, conducting the context switch without further cache write-back to memory.

20 5. A method as claimed in Claim 3, and including the step of conducting a cached data write-back to memory at the time of requiring a context switch if the said flag has not been identified as set.

25 6. A computer implemented system including a processor and cache memory arranged to receive operating system instructions for context switching between processes, and including control means for writing back cache data to memory means during processor idle cycle at completion of a process, and prior to initiation of the context switch.

30 7. A system as claimed in Claim 6, and arranged to employ a processor sleep mode to write cache data back to memory automatically during a sleep period.

8. A system as claimed in Claim 6, and including a status register having a flag for initiating data cache write-back during a CPU sleep operation.

5 9. A system as claimed in Claim 6, wherein the operating system is arranged with an additional sleep code instruction for the data cache write-back

10 10. A system as claimed in Claim 9, wherein the sleep code of the processor includes an additional data cache write-back instruction.

15 11. A computer program product having computer program instructions and arranged for controlling context switching between processors in a computer operating system so as to write cached data back to memory means during processor idle cycles at completion of a process and prior to initiation of the context switch.

20 12. A computer program product as claimed in Claim 11, and including instructions for controlling to execution of a method as claimed in any one or more of Claims 2 to 5.

## ABSTRACT

**METHOD AND APPARATUS FOR CONTEXT SWITCHING  
IN COMPUTER OPERATING SYSTEMS**

5

The present invention provides for a method and apparatus for a computer implemented system including a processor and cache memory arranged to receive operating system instructions for context switching between processes, and including control means for writing back cache data to memory means during processor idle cycle at completion of a process, and prior to initiation of the context switch so as to enhance the operating speed since, at the time of initiating the context switch, the cached data has already been written back to memory.

10

1/3

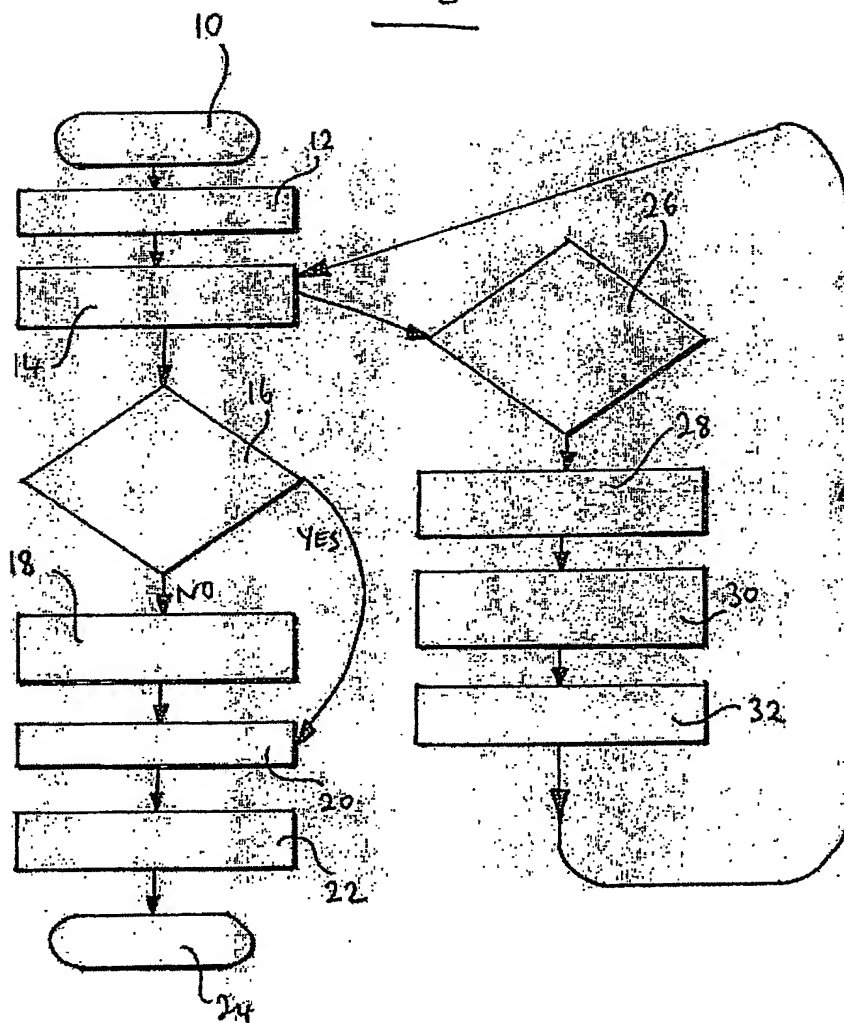


FIG. 1

2/3

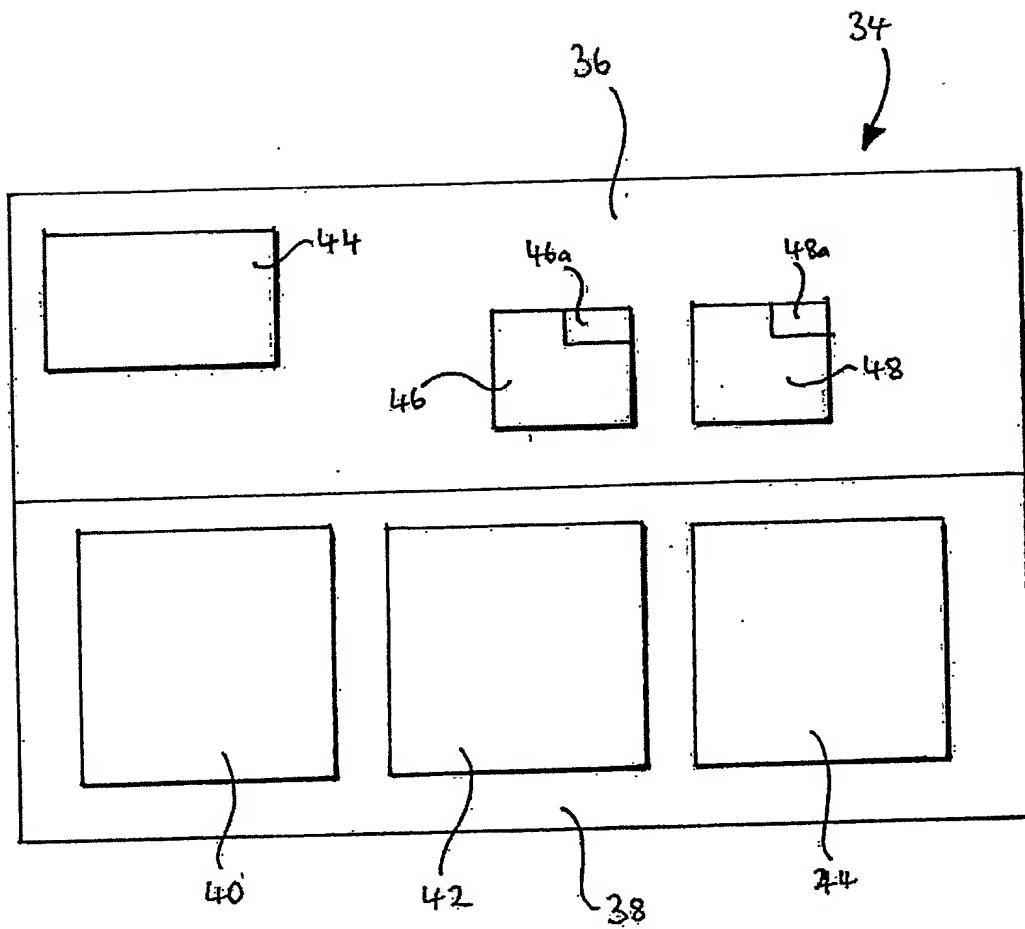


FIG. 2

3/3

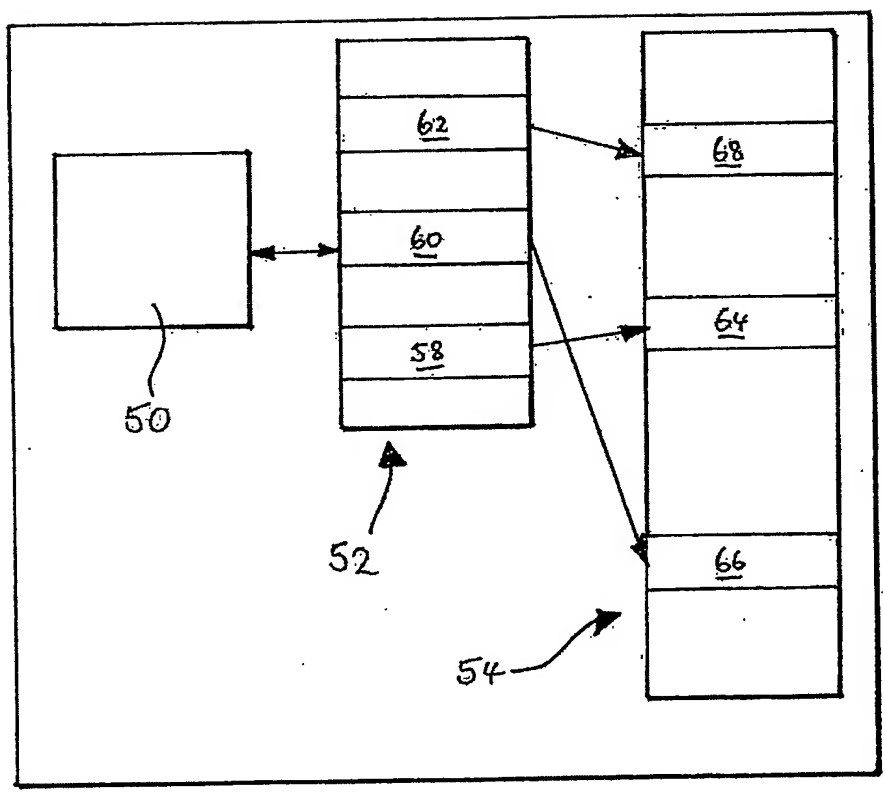


FIG. 3